# VisualPet Message Router

**INTRODUCTION DOCUMENTATION AND SAMPLE APPLICATION**

SEA AIR & LAND COMMUNICATIONS LIMITED, CHRISTCHURCH, NEW ZEALAND

## VisualPet Message Router

VisualPet is Salcom designed message router (The "Visual" relates to the configuration controls being like VisualBasic, VisualC++, The Pet relates to the first protocol designed, industry standard PET/TAP ) VisualPet configurations allow the routing of messages from multiple input sources to multiple output destinations, ie:

Inputs: (examples only)

| | | | |
|---|---|---|---|
| Salcom 11-36 Fone Page | PET / TAP | TNPP | VPGen |
| Email ( SMTP / POP ) | Internet | Selcall | X25 Pacnet |

Outputs: (examples only)

| | | | |
|---|---|---|---|
| Salcom 11-62 Paging Transmitter | PET / TAP | Selcall | Email (SMTP / POP ) |
| VPGen | Internet | X25 Pacnet | TNPP |

The VPGen protocol allows scripted configuration of just about any industry protocol. Salcom can create protocols to suit any application where a message or packet of information needs to be monitored, routed and delivered. Also protocols can be configured to operate using RS232 serial or TCP/IP. Eg: TNPP or PET over IP for long distance or network message delivery connections

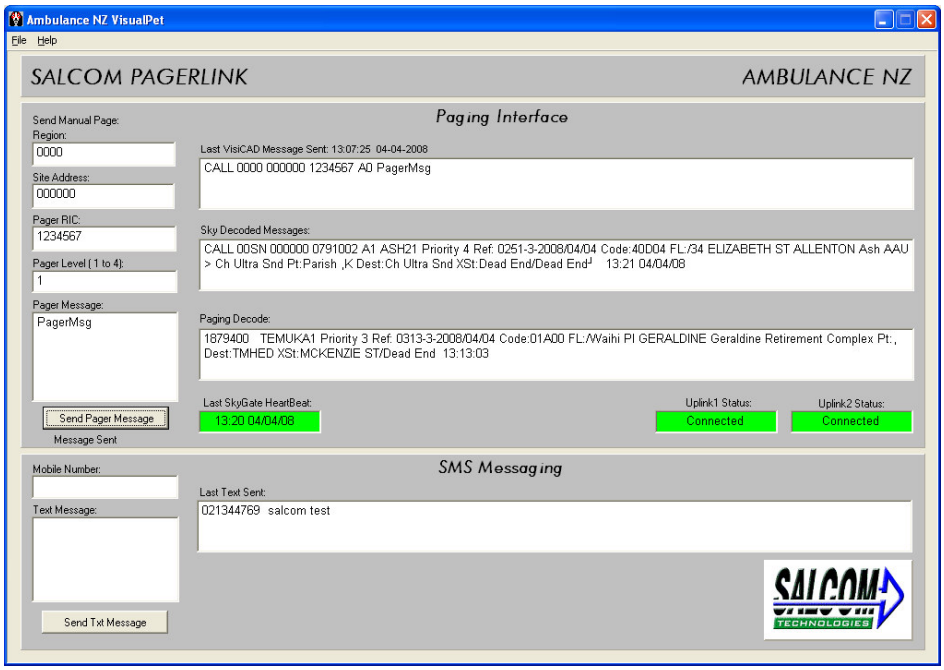
Each message route allows delivery to multiple destinations from a single source message. The VisualPet router can translate contact, message type, and specific protocol information for any inbound and outbound message. For example an inbound PET message can be forwarded onto a pager through an 11-62 with a copy sent via email and TNPP to another paging vendor.

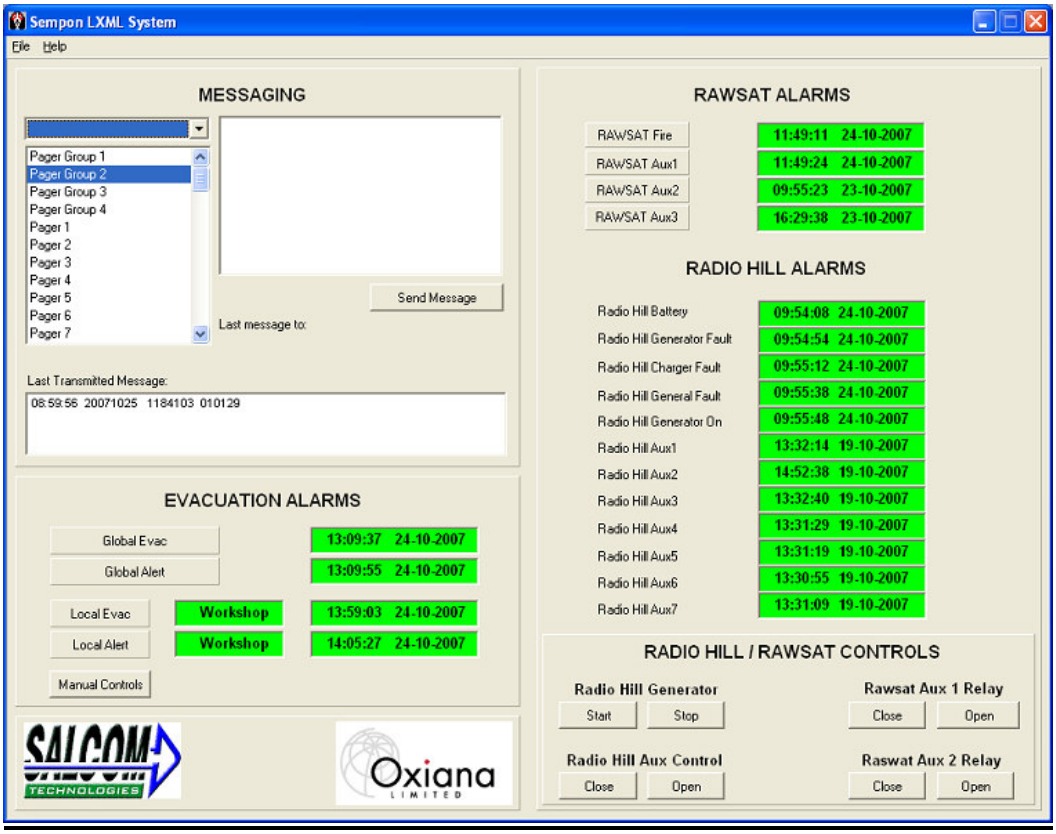
Multiple inbound and outbound messages can be handled at one time, with complete logging of all message transactions. A message delivery failure can generate an alarm route will contain the connection, contact and message that failed. Contacts within the router are easily added, modified and deleted.

If a new protocol is required Salcom can create a VPGen script or add a new protocol to the VisualPet protocol library. Recent VPGen scripts have communicated with GSM cellular modems or connecting via the internet to SMS gateways using SNPP or HTTP POST for txt delivery.

New Protocols have been created for specific customers to decode serial data presented from satellite or paging receivers.

The Visual controls of VisualPet allow users to create a visual package that suits the deployment requirements of the application. Examples below are Salcom deployed systems for multiple source and destination routers or alarm and messaging systems.



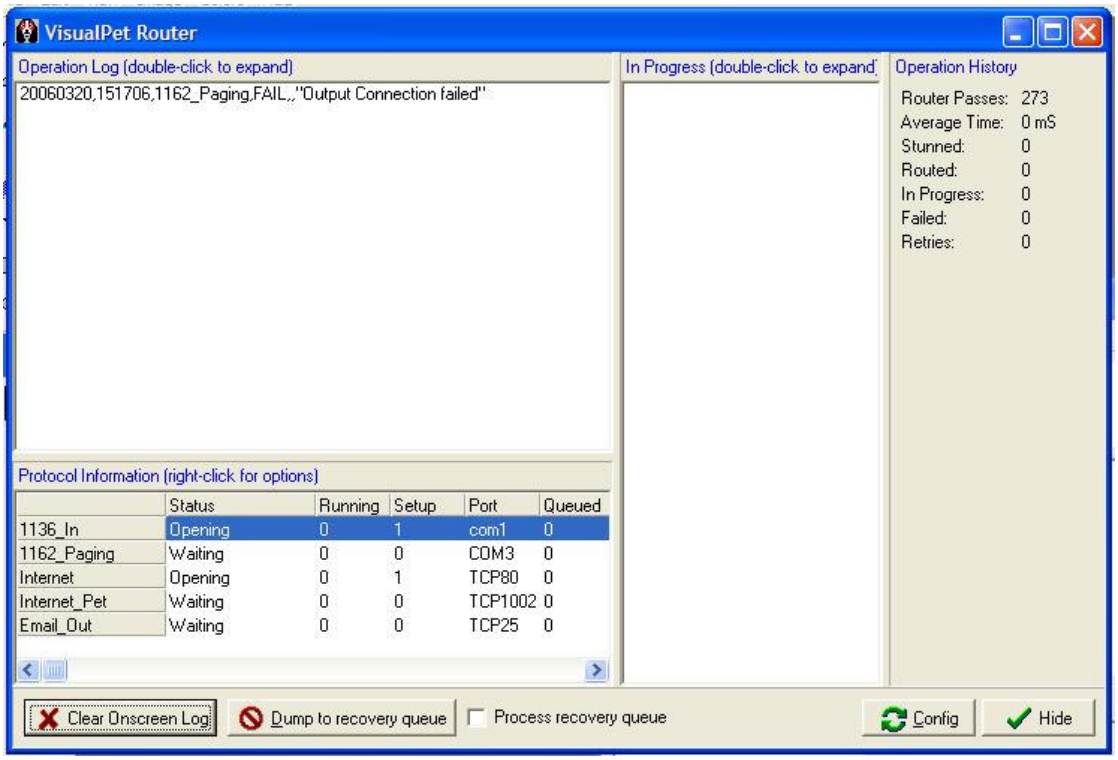




The inbuilt programming environment within VisualPet allows application changes/designs without the need to re-write any application source code.

## Message Router

The message router allow users to see the transactions and messages being accepted and delivered by VisualPet.

To view the router screen use "File > View Router", this will display the screen below:




The router screen displays information about each message into and out of VisualPet.

The top left screen shows all messages that have been recently routed.

The bottom left shows the status and port settings for all running protocols. (nb: you may see protocols here that were not available in the delivery options for a contact, this is because some of the protocols shown here are input only).

The "Clear Onscreen Log" button will clean the operation log screen. The "Dump to recovery queue" can be used to save current messages within the router when a delivery destination has failed, this allows the destination connection to be repaired and then the "Process recovery queue" tick box can be used to pull the queued messages back into the router for delivery.

The "Operation History" top right will show how many messages have been successfully routed, are in progress or have failed.

The "Hide" button will close the router window and return to the Main window.

## VisualPet Installation

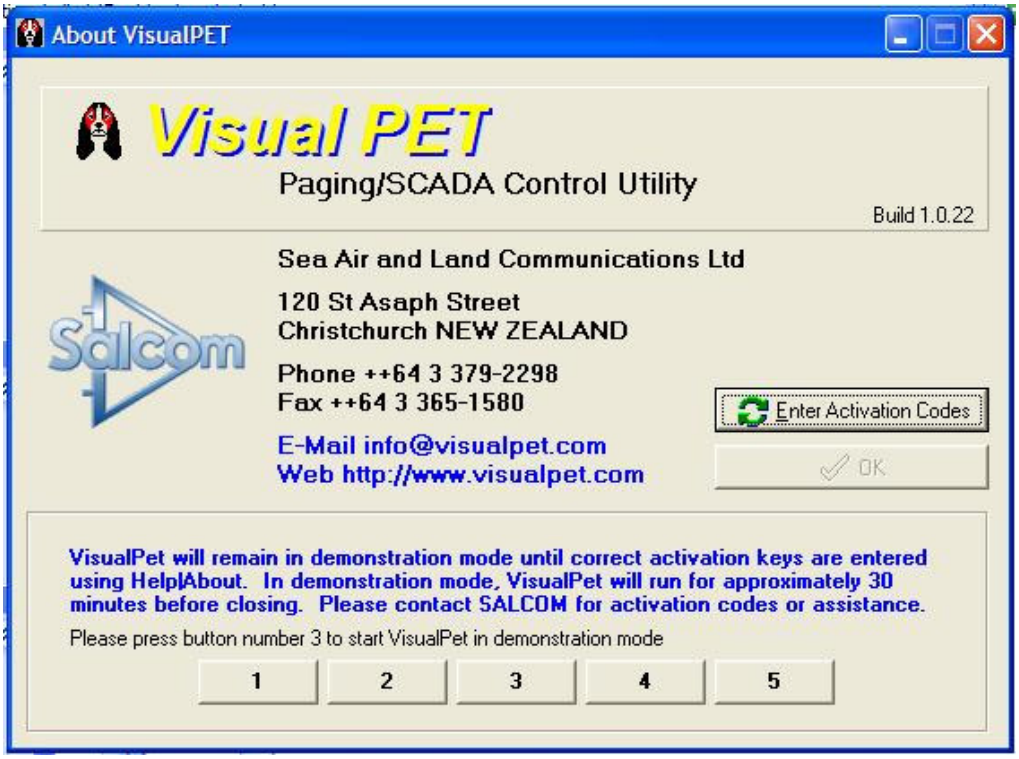
VisualPet will be shipped on a CD Rom pre-configured to specific requirements thus allowing a simple process to install and setup.

VisualPet is a Windows compatible program which can be used on Win98, NT, Win2000, XP operating systems.

Simply run the Setup.exe file supplied and follow the onscreen instructions

## VisualPet Activation

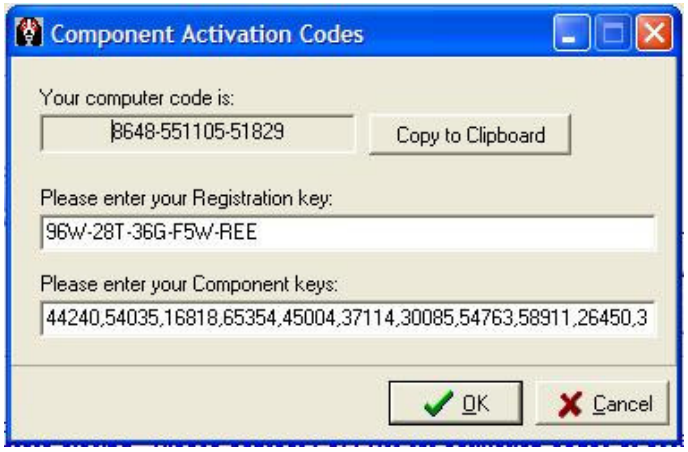
The first time you run VisualPet an "About" screen will load asking for a button to be pressed to allow VisualPet to be used for a period of 30 minutes in a demonstration mode. After VisualPet has been activated this screen will no longer appear and VisualPet will run continuously.


About VisualPET
Visual PET
Paging/SCADA Control Utility
Build 1.0.22
Sea Air and Land Communications Ltd
120 St Asaph Street
Christchurch NEW ZEALAND
Phone ++64 3 379-2298
Fax ++64 3 365-1580
E-Mail info@visualpet.com
Web http://www.visualpet.com
Enter Activation Codes
OK
VisualPet will remain in demonstration mode until correct activation keys are entered using Help|About. In demonstration mode, VisualPet will run for approximately 30 minutes before closing. Please contact SALCOM for activation codes or assistance.
Please press button number 3 to start VisualPet in demonstration mode
1
2
3
4
5

VisualPet loses no functionality running in demonstration mode and will route messages to and from all sources and destinations but will shut down automatically after 30 minutes operation.

Use the "Enter Activation Codes" button to start the process of registration and activation of VisualPet. Remember that VisualPet must be installed on the PC that will be used to run the software as transferring to another PC after activation will mean the activations codes will become invalid.

The "Enter Activation Codes" button will open a separate screen:




The number in the "Your computer code is:" box will need to be copied and sent via email to: sales@salcom.co.nz. This number is unique to the computer that the software was installed on therefore even if the program is activated and then copied to another PC VisualPet will again require activation.

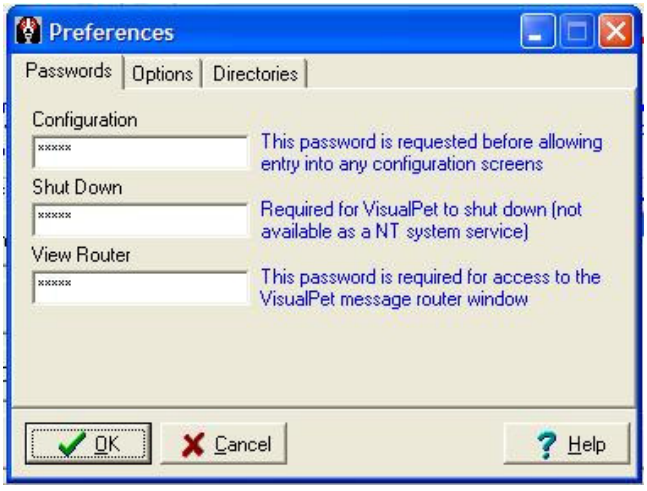
After receiving the computer code Salcom will reply with a Registration Key and Component Keys which can be copied into two appropriate boxes, "OK" will then activate VisualPet. VisualPet will need to be shut down and restarted for the activation to be complete.

**Shutting VisualPet Down**

When VisualPet needs to be shut down the "File > Exit" will prompt with a password. This stops VisualPet being closed accidentally, the default password is "close"

To change the password navigate through "File > Configure" then "Configure > Preferences". The window will allow passwords to be set for Configuration, Shut Down and View Router.

The Configuration and View router passwords will protect VisualPet configs from being modified by unauthorised users.

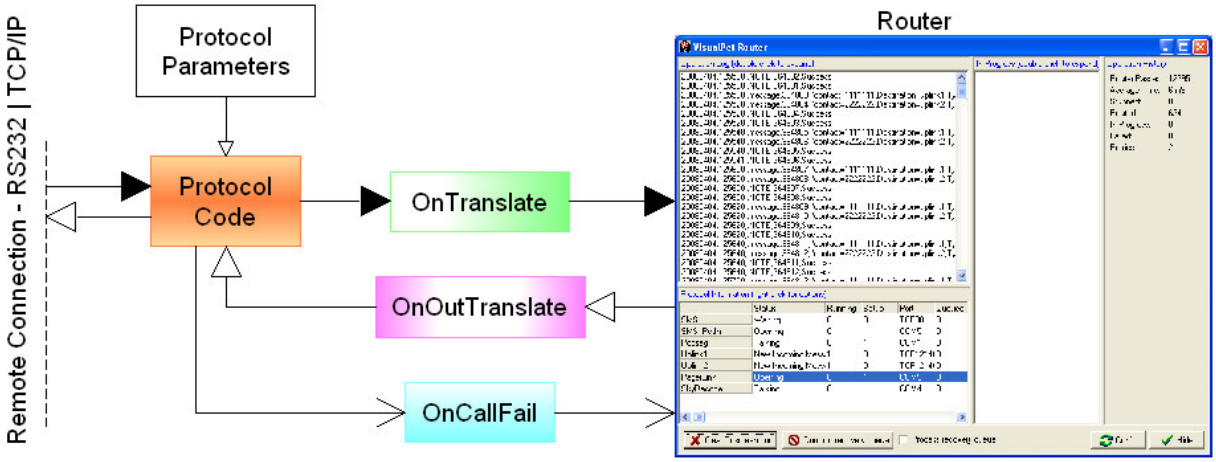


## Protocols

VisualPet has options for multiple protocols allowing different types of connection for delivery and reception of messages into and out of the router. Because VisualPet is a multithreaded application each protocol can handle connections and data totally independently of each other, all at the same time.

Each protocol has a range of configurable settings for the type of communication required to an external device. Protocols can be configured to operate on serial com ports or TCP/IP ports with configurable parameters such as port number, connection speed etc..

Each protocol also has code space for handling possible inbound, outbound or failed messages.




The diagram above explains how a protocol communicates between a remote connection and the VisualPet Router. Text wrapped in [ ] refers, within the explanation below, to the Protocol sections diagram above.

**Incoming Messages:**

An incoming message from a remote connection is first handled by the specific [Protocol Code]. The code has parameters that are defined by the protocols specification (ie: PET/TAP has a specification on how it communicates) plus user configurable [Protocol Parameters] such as RS232 or TCP/IP, Baud Rate etc…

Once the [Protocol Code] has received a valid message this message is passed to [OnTranslate] for any user configurable processing before passing into the VisualPet router. The [OnTranslate] is a code space which using VisualPet's script language (TASP) allows scope for almost infinite processing possibilities.

Examples are, logging the message to file, displaying the message on screen, creating billing information message counts, re-routing the message directly to another protocol (or multiple protocols), modifying contact or message information, adding or removing information.

**Outgoing Messages:**

A message in the router queued for delivery to a [Remote Connection] is firstly passed to the [OnOutTranslate] for any user configurable processing before passing to the [Protocol Code] and finally delivered to the [Remote Connection]. The [OnOutTranslate] is a code space which using VisualPet's script language (TASP) allows scope for almost infinite processing possibilities.

Examples are (as above) the same as [OnTranslate] information.

**Failed Messages:**

If a message passed to a protocol is not able to be delivered, Eg: The remote connection in broken, network down or RS232 cable not fitted, the [Protocol Code] will use counters and timers configured in the [Protocol Parameters] to retry message delivery.

Once these retries have failed the [Protocol Code] will pass the message to the [OnCallFail] code space where a user can configure a script to handle the failed message.

Examples maybe to log the message to file, send a specific contact (eg: Network Administrator) a call fail notification plus a copy of the message.

**VPGen Protocol**

Salcom's VPGen Protocol was created to allow the user access to the [Protocol Code] which allows creation of a script to provide the direct communications between the VisualPet Router and the Remote Connection. This is useful when a standard VisualPet protocol does not meet the requirement for communications to a remote connection.

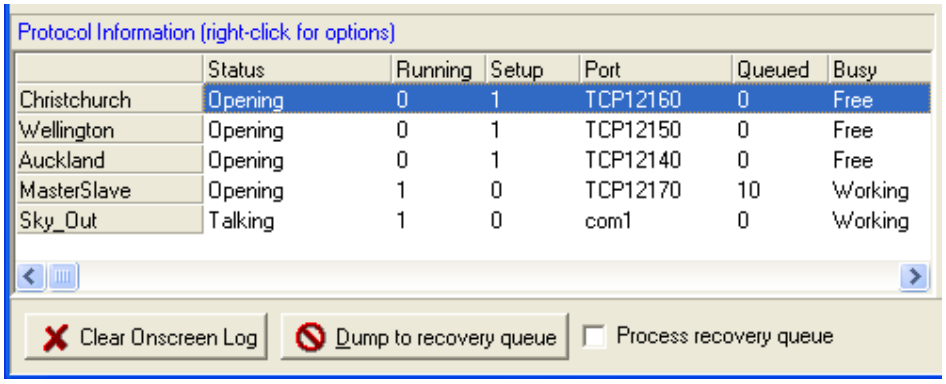
Some protocols are single direction only such as Pet In or duplex like TNPP and will therefore only have code space for the communication directions available ie: [OnTranslate] for Pet In.

To view setting and configurations for a protocol use File > View Router (from the main window) to open the VisualPet Router.

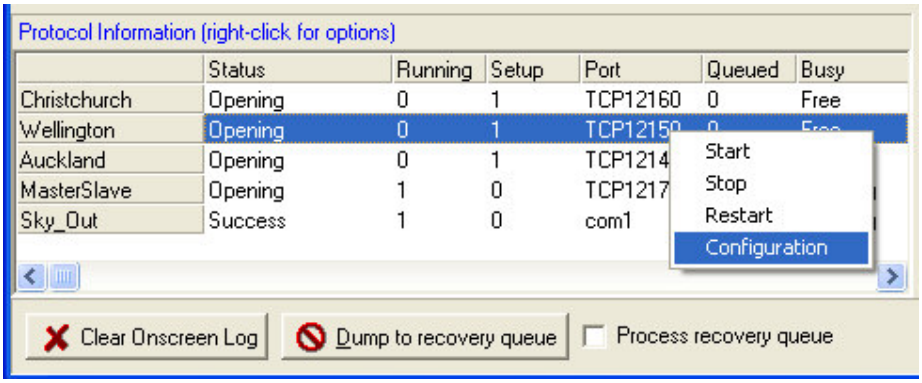
The lower left window of the router called "Protocol Information" contains the list of running protocols.

The Protocol Information displays the Protocol Name (eg: Sky_Out) and also shows the status of the port (ie: If the port is open, connected, receiving new message). Also shown is what type of connection the port is configured as.
ie:
Christchurch = TCP12160, configured as TCP/IP on port 12160.
Sky_Out = COM1, configured as RS232 serial port on Com 1.

Other information available is the number of messages queued and if the port is "busy" or "free".


Protocol Information (right-click for options)
Status Running Setup Port Queued Busy
Christchurch Opening 0 1 TCP12160 0 Free
Wellington Opening 0 1 TCP12150 0 Free
Auckland Opening 0 1 TCP12140 0 Free
MasterSlave Opening 1 0 TCP12170 10 Working
Sky_Out Talking 1 0 com1 0 Working
Clear Onscreen Log
Dump to recovery queue
Process recovery queue

To change, view, stop, start or restart a Protocol, highlighting a particular protocol, then right mouse clicking will display a menu of options:




**Start** will start a protocol if for some reason is was stopped

**Stop** will stop the protocol running and remove the connection to either a TCP/IP Port or RS232 Com Port

**Restart** will firstly Stop the protocol, freeing the port, and the Start the protocol again taking control of the configured port.

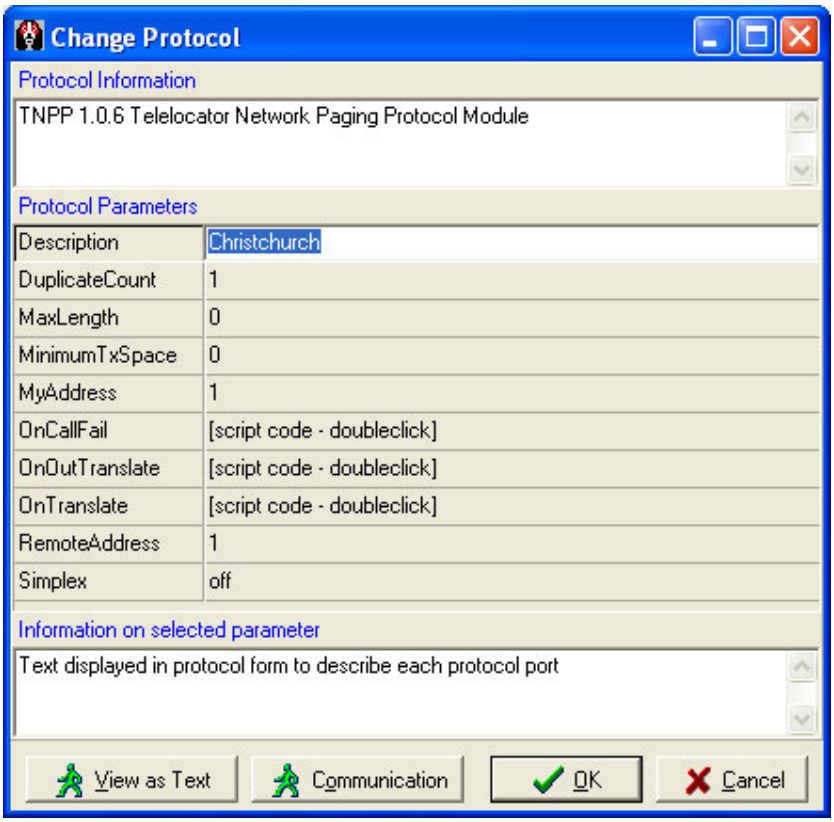
**Configuration** will open the window for configuring the selected protocol, allowing changes, or viewing, to the setup or scripts if required.

## Protocol Configuration

VisualPet will normally be shipped with the customers protocols pre-configured ready for use, except for the actual setting of RS232 com ports and TCP/IP addresses all aspects of the configuration should be ready to use.

To configure an installed protocol highlight the protocol from the router screen then right click and use the Configuration option to view the window below:

The example below is the Change Protocol window for the TNPP protocol.




**Description** is the name used to identify the protocol. Eg: Christchurch.

**DuplicateCount** allows multiple copies of a message to be sent, default = 1.

**MaxLength** sets maximum numbers of characters in a message, default = 0, no maximum.

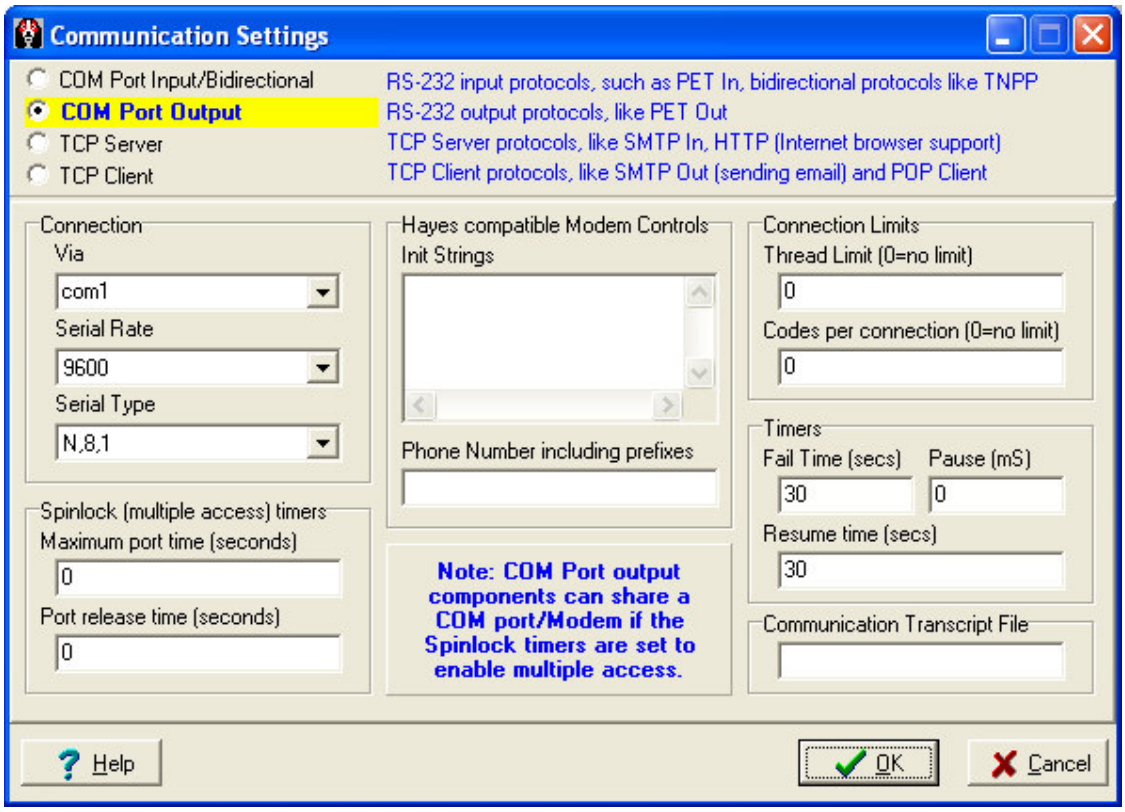
**MinimumTxSpace** set a time between queued message in milliseconds, set to 100ms.

**MyAddress** is used for TNPP networking, set to 1.

**OnCallFail** allows code for handling a failed message delivery.

**OnOutTranslate** The code space here is for handling messages *before* they are sent.

**OnTranslate** The code space here is for handling *received* messages *from* a remote device/terminal at the other end of a communications link.



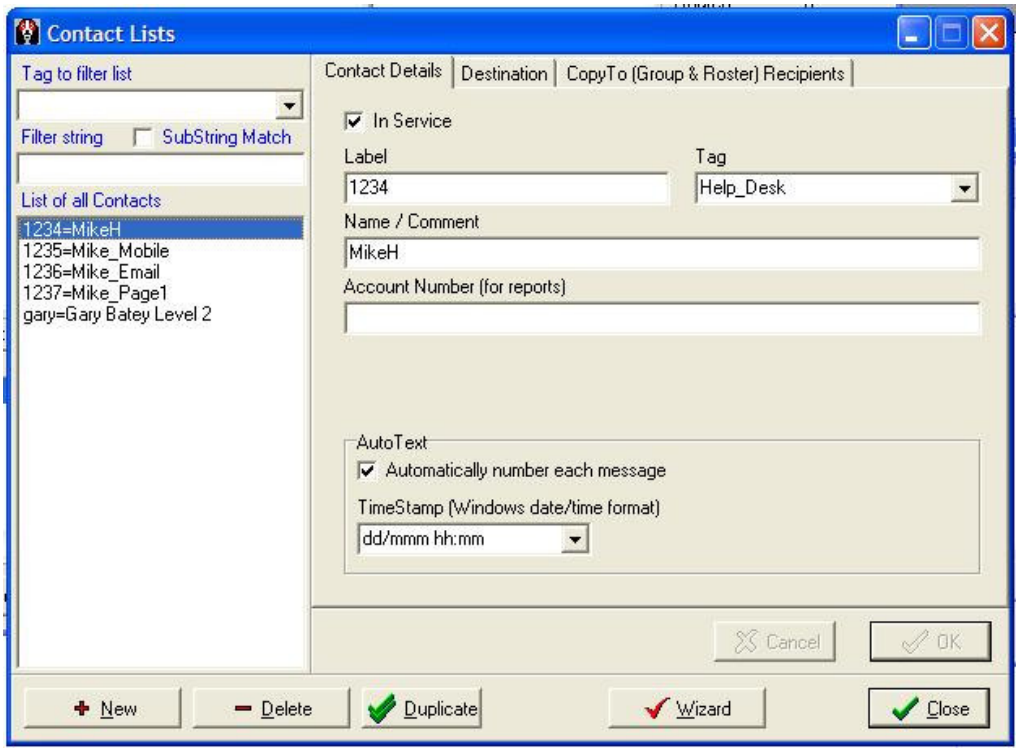

Only the settings under the "Connection" window are likely to need changing. Select the correct Com Port from the drop menu and change the serial rate and type if required.

A log file can be created for all data transactions through the port by typing a suitable file name in the "Communication Transcript File" box.

NB: When using the 11-36 Telephone interface set the serial port to 9600,N,8,2

## Contacts

To add a contact and select the destination for message delivery select "File > Configure". Then from the menu shown click the button labeled "Contact Lists". The screen below will appear and from here contacts can be added, modified and deleted:




The opening/default tab is "Contact Details"

The "List of all Contacts" shows current contacts within VisualPet. The highlighted contact (1234=MikeH) is therefore the contact whose details are shown under the "Contact Details" tab.

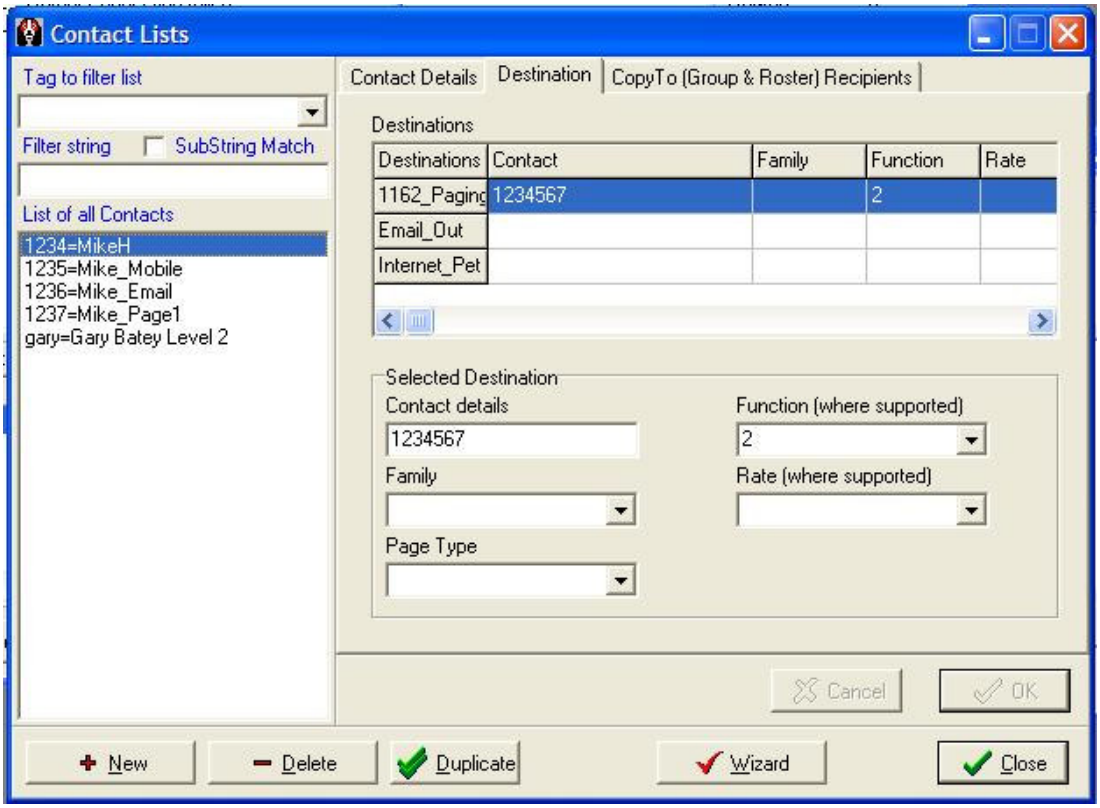
The "In Service" box allows the option of taking a contact out of service without losing the detail.

The "Label" is the contacts ID within the router. Eg: If you have an 11-36 Fone Page connected then the only suitable contact labels will be four digit numbers because a touch tone phone doesn't allow alpha characters to be entered, most TAP and TNPP type connections into the router use numeric ID's for contacts. Email in for example could use alpha characters.

The "Name / Comment" field is where you can attach a more suitable name or comment about a contact. This name will be what appears in the Main window for selection and sending messages directly from VisualPet.

If multiple contacts are to be added and they have different groupings then the "Tag" field can be used to allow easy viewing from the Main window. Eg: Tags likes Sales, Admin, Technical can be selected from the main window to view only the Sales Team.
The "AutoText" and "TimeStamp" allow each outbound message to be numbered and time stamped.

The "Destination" tab is used to select where a message to the contact is to be delivered:




The "Destinations" box will show a list of installed protocols for message delivery.

The above example has three destination options with only the 1162_Paging option having a contact entered. The Salcom 11-62 requires a seven digit RIC code (1234567) in the "Contact details" and a "Function" ( 2 ) for the alert level on the pager.

If for example the message delivery was to include an email address then by selecting the "Email_Out" protocol and entering a valid email address (mikeh@salcom.co.nz) in the "Contact details" every time a message for 1234=MikeH (as seen on the "List of all Contacts") was sent the pager and email would both receive it.

No syntax checking is done on the value for each contacts delivery details, so if an incorrect value was entered ( eg: mikeh#salcom.co.nz ) then the message would fail when the "Email_Out" protocol tried to deliver it.

Once all contact details are entered a click on the "OK" will save the settings and the "Close" button bottom right will return to the Main screen.

**Examples for VisualPet:**

**Inhouse Paging Systems**

VisualPet has been used in hospitals, hotels and other businesses requiring in-house paging systems. VisualPet can provide multiple methods for sending and receiving messages, for example VisualPet resides within the building on the local area network (LAN) and has a desktop configuration which allows a user to select a Pager (contact) and type a message then send it. This desktop is also accessible from any PC on the LAN by simply using an Internet browser.

The message for sending can then be passed to a Salcom 12-62 Paging Transmitter (via a local serial port or via the LAN) for transmission. A copy of the message could also be delivered to a Telco as an SMS and/or emailed.

**Legacy Support Systems**

VisualPet has enabled many companies with existing software packages to add modern solutions for message handling. Many companies have large and expensive software packages which have prohibitive upgrade costs for adding new technology, this is where VisualPet excels.

**Examples:**

**Call CentreTAS (Telephone Answering Service)**
Call Centre handling thousands of inbound calls for multiple clients each month with an existing system that only supported PET/TAP for outbound message delivery. The Telco's changed the model for delivering txt and paging messages from PET/TAP to an HTTP POST (Web based SMS Service) system.

The Call Centre hardware/software was not upgradable to this technology so VisualPet was employed to receive messages from the TAS and convert these messages into the HTTP POST for delivery.

**Automated Paging System for exchange rates and market information.**
This software system received continuous financial market information and would send the data to remote clients using a Wide Area Paging network.

The data was delivered to the Paging network using an X.25 connection (PACNET). The X.25 network was eventually decommissioned so VisualPet was used to accept data as X.25 and pass it to the Paging network using email.

## Sample Application

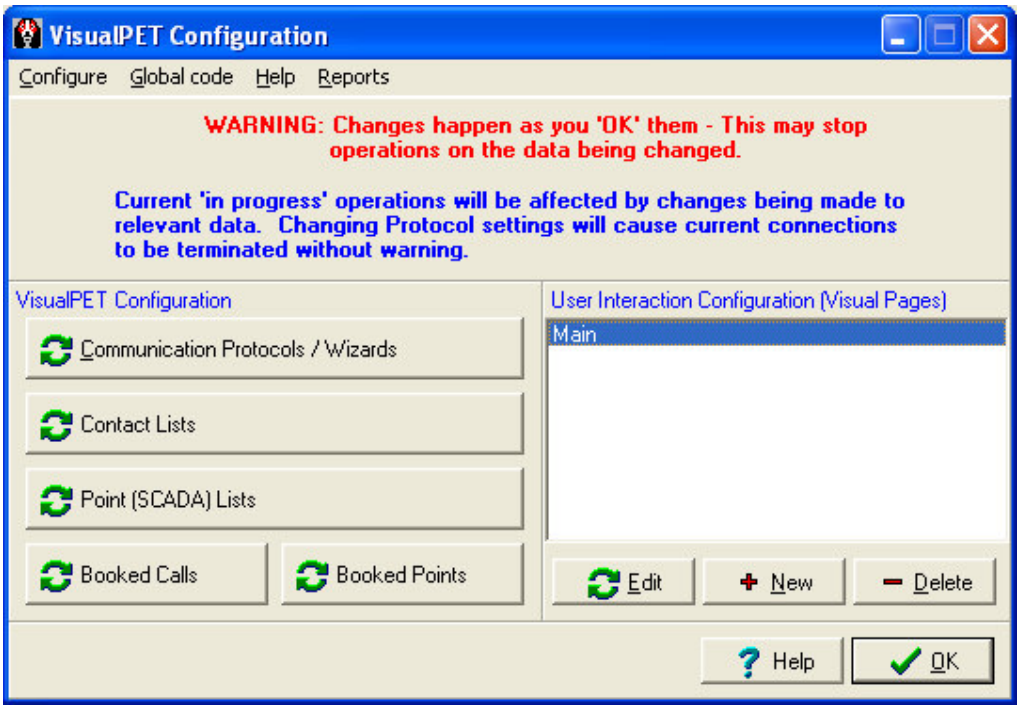
Firstly download the VisualPet sample system (http://vprouter.salcom.co.nz/Visualpet_Demo.zip). Once you have unzipped the setup files run the vpet_setup.exe file to install.

With VisualPet installed double click the VisualPet icon on the desk top to start the application. As detailed above VisualPet requires activation but at this stage it is not required to work with the software as a demonstration, so click the required 1,2,3,4 or 5 button the begin using.

From here we are going to create a simple in-house paging system to connect to a Salcom 12-62 for sending POCSAG paging messages.

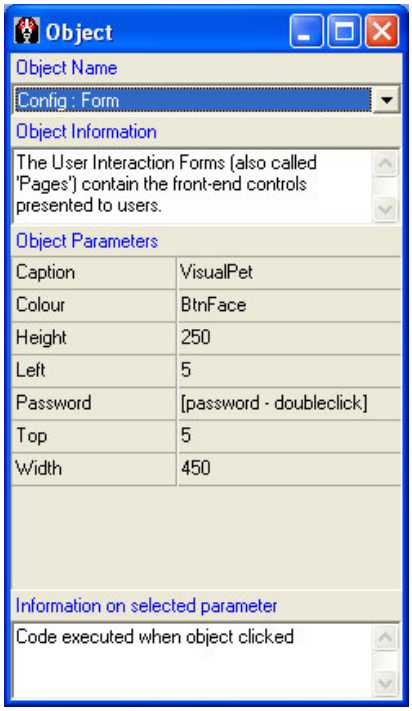
### Step 1: Configuring Controls.

Select File > Configure. This will summon VisualPet's configuration interface. Inside the configuration interface is a window called User Interaction Configuration (Visual Pages) where you can see a page called "Main". Highlight "Main" and click the Edit button below.



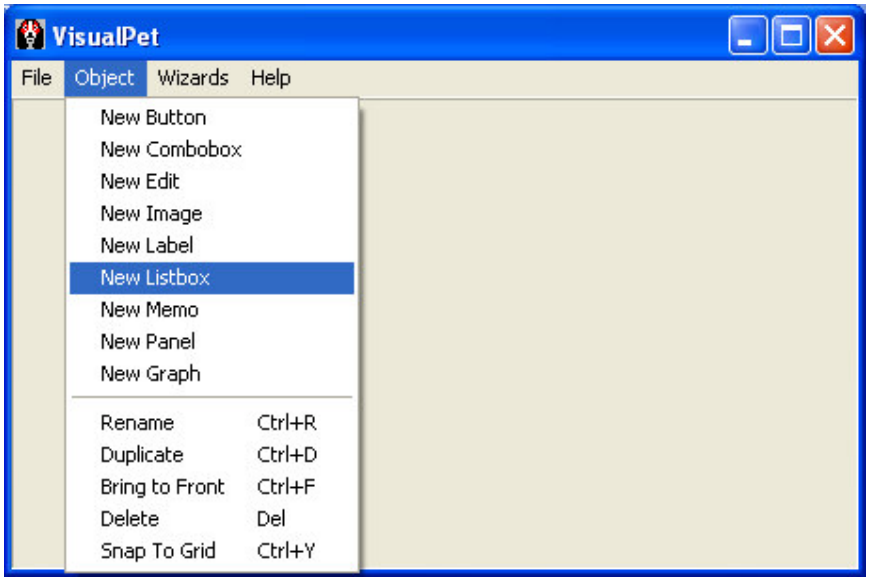
A form (probably still blank at this stage) with an attached "Object Inspector" will appear. This is where we can place and arrange the controls (visual objects) that a user can interact with.

Object Inspector:

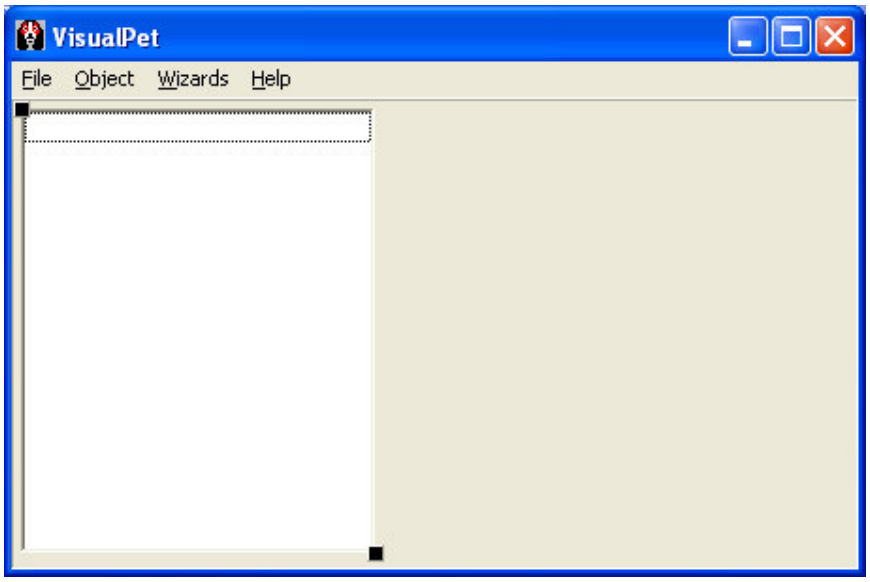



Resize the form - either by using the mouse to drag the bottom-right corner, or by simply changing the values in the Object inspector. We want a Height of about 250 and a Width of about 450.

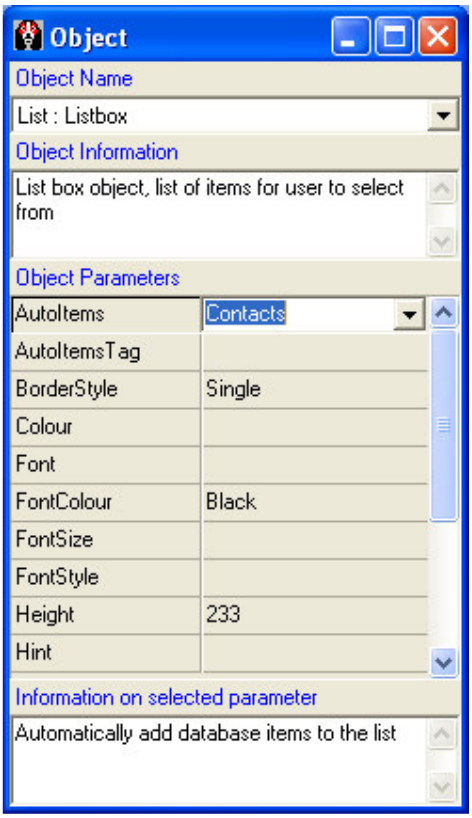
At this stage, we want only three controls. Select Object > New Listbox to create our pager selection box, and name it List (instead of the default Listbox1).



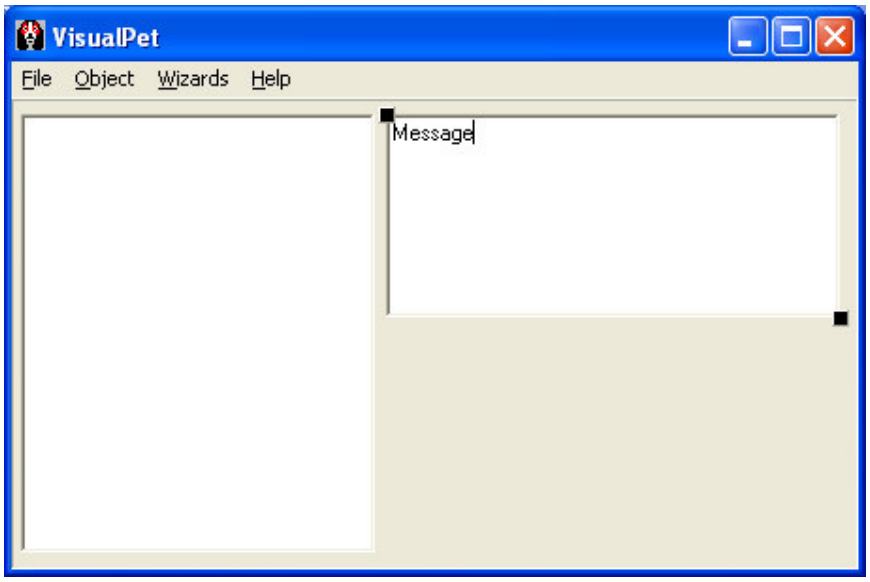
The listbox will appear in the top left corner of the form, and we'll leave it there for now, however, drag the bottom-right resizing point (small black box) down to the bottom of the form to make the listbox almost as high as the form.



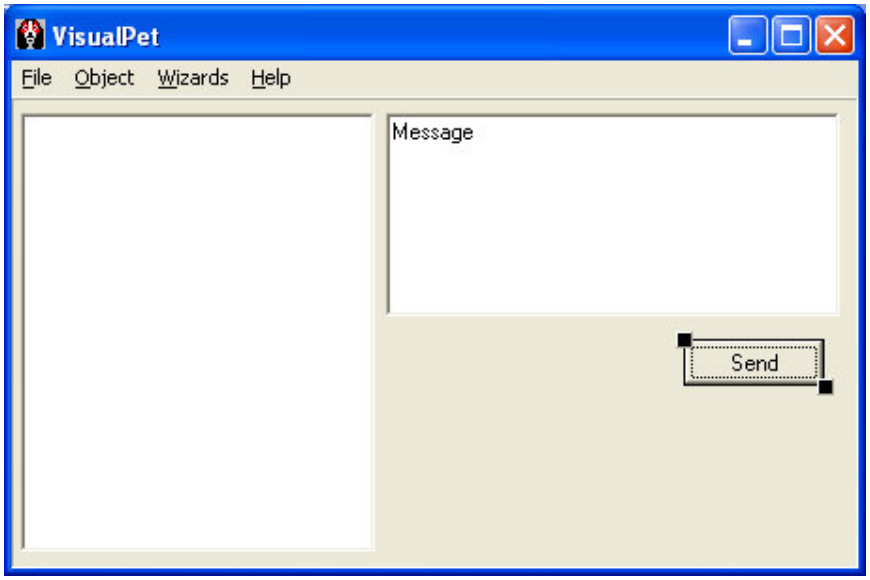

We want this List to be populated by our contacts when we add them, so in the object inspector the command at the top of the list called AutoItems change the value to Contacts



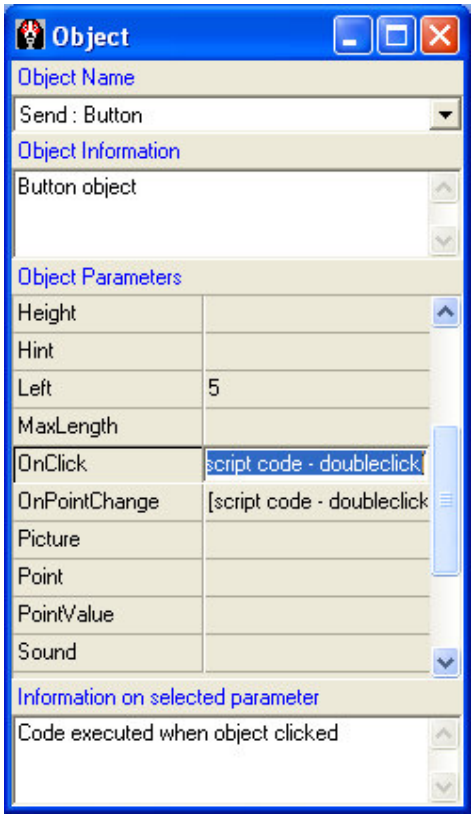
Next, we want a message entry window. A free-form text entry box is called a 'memo'. Object > New Memo gets us one of these, call it "Message" and drag it (using the mouse) to a spot just right of the listbox. Note: Object > Snap To Grid makes sizing and lining up controls much easier.




The third control we want is a button. Object > New Button, call it Send, and move it to just below the memo.

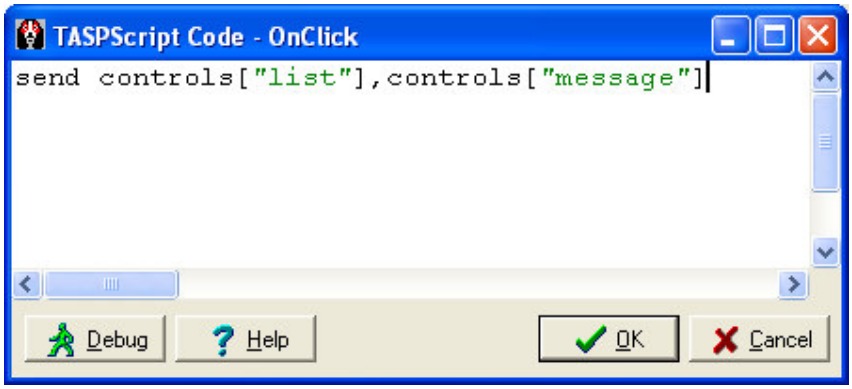


We want the button to send the message to the contact in our listbox, so we have to write a line of code for the button. Double-click in the object inspector on the [code] to the right of OnClick.




This will summon a window titled "TASPScript Code" - OnClick. This code will be executed whenever the user clicks the Send button. Add the text below to the TASPScript.

**send controls["list"],controls["message"]**




The send command initiates a paging call, controls["list"] is the contact selected from the listbox (we'll fill that out in a moment) and controls["message"] is the text in the memo. Press OK to save your first piece of script code.

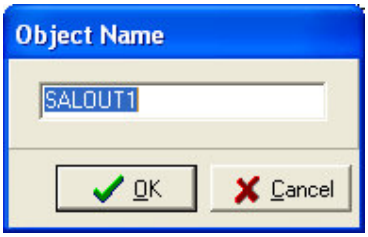
We have now completed a simple form for sending messages use the X button on the top of the form to close it and bring us back to the configuration window.

**Step 2: Configuring Protocols.**

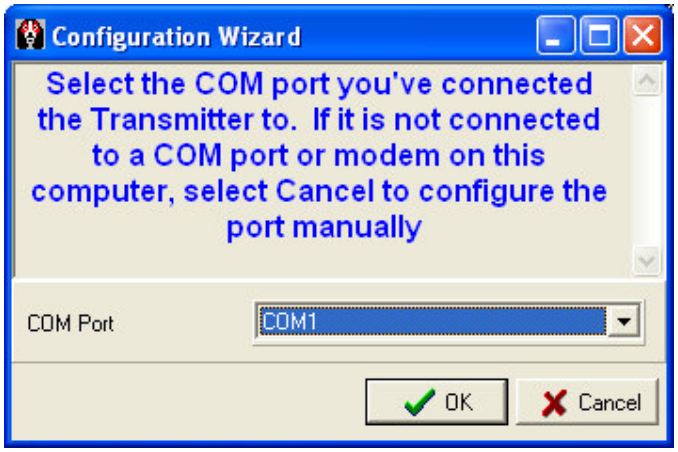
The only worldly interaction we want in this application is a paging connection to a local transmitter for our five pagers.

In the Configuration Window click the button "Communication Protocols / Wizards".

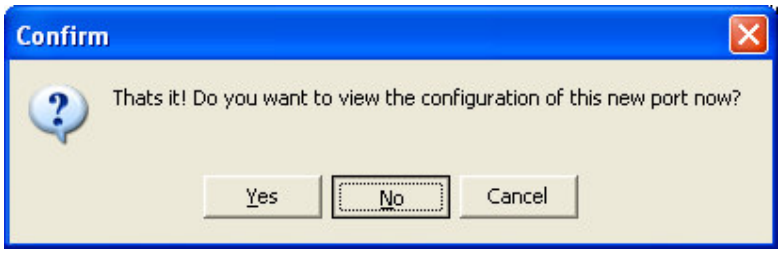
In the Available Protocols, Double Click on "VPSALOUT" protocol and call it "SalcomOut1".




Leave the default of COM1 (or change it to a connected ComPort).


Configuration Wizard
Select the COM port you've connected the Transmitter to. If it is not connected to a COM port or modem on this computer, select Cancel to configure the port manually
COM Port
COM1
OK
Cancel

Click "OK" to save these settings, Then click No to see the configuration.



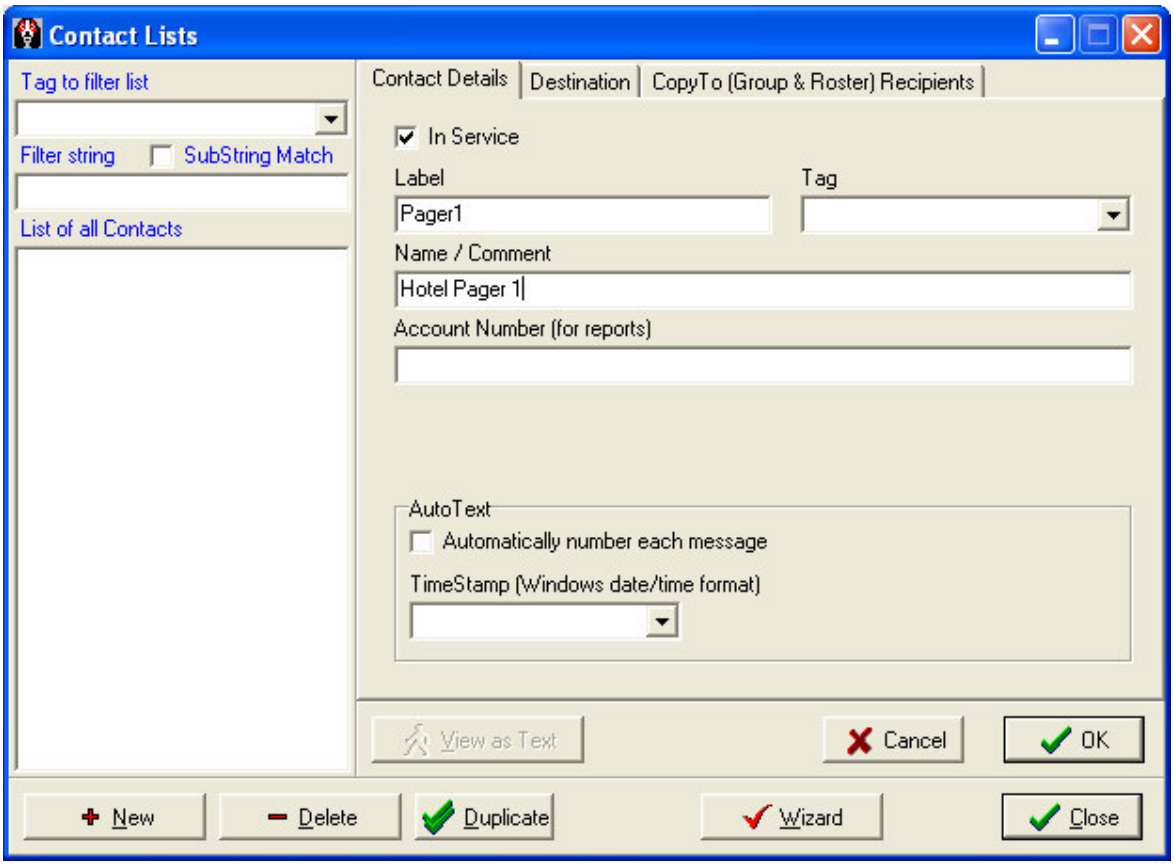

This returns us to the main configuration window again.

**Step 3: Configuring Pagers.**

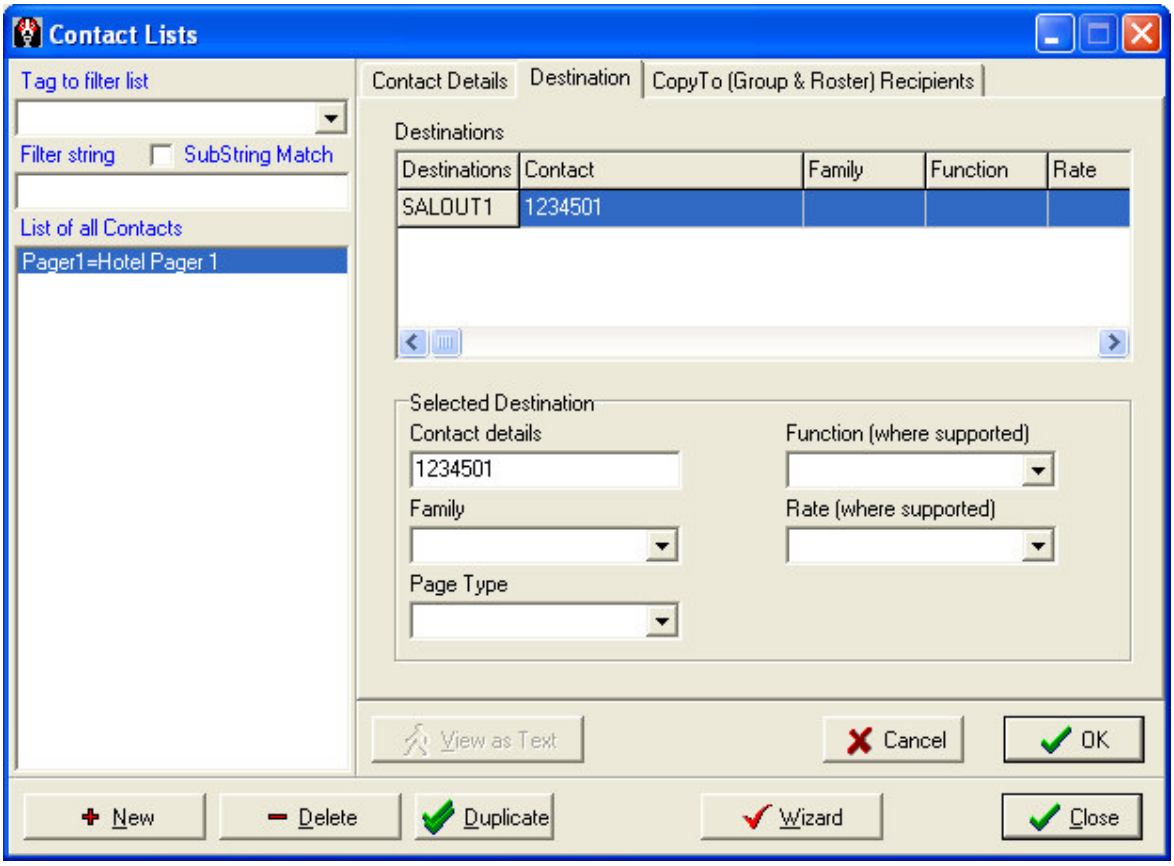
Our application requires five pagers.

Click "Contact Lists" to summon the contact lists editor.

Press "+ New" to create a blank pager. Enter the Label Pager1, and the name "Hotel Pager 1".

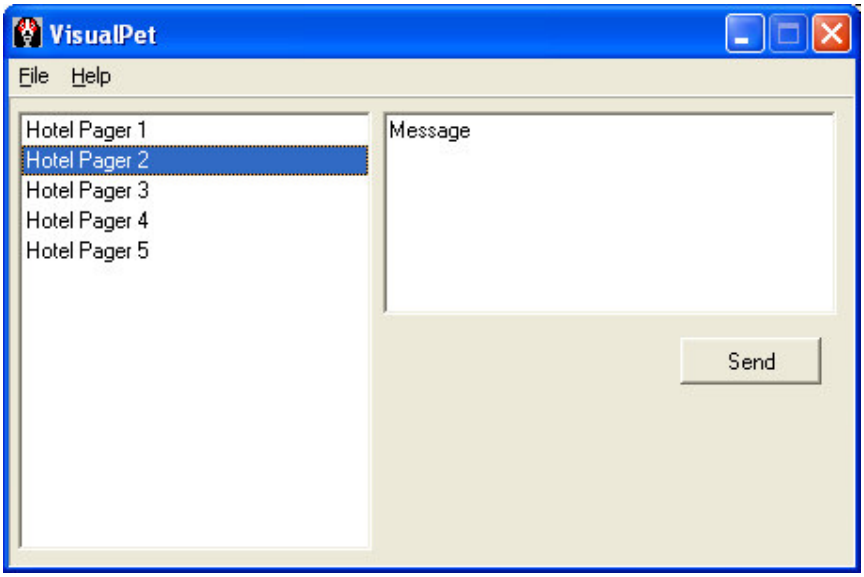


Click on the Destination tab, and enter 1234501 in the contact details area for SalcomOut1 (there should be only one destination at the moment).




Press "OK" to save this pager, then repeat this sequence for Pagers 2 through 5 (contacts 1234502 through 1234505).

Press "Close" to finish, then "OK" in the configuration window to return to VisualPet's main user interface. Notice how the list box is now full with pagers?

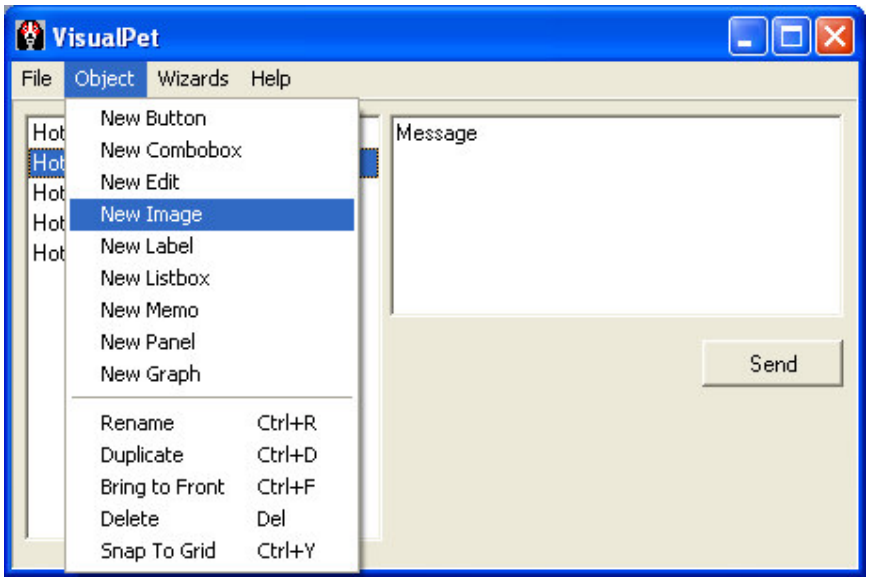


*Thats it! You've configured VisualPet with the five pagers and one destination.*

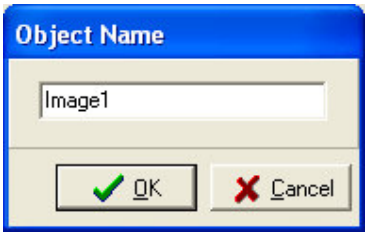
So from here if you have a 12-62 paging Transmitter connected to the configured com port, type a new message in the Message window and click the Send button and you've just sent a message using VisualPet.

**Simple Modifications**

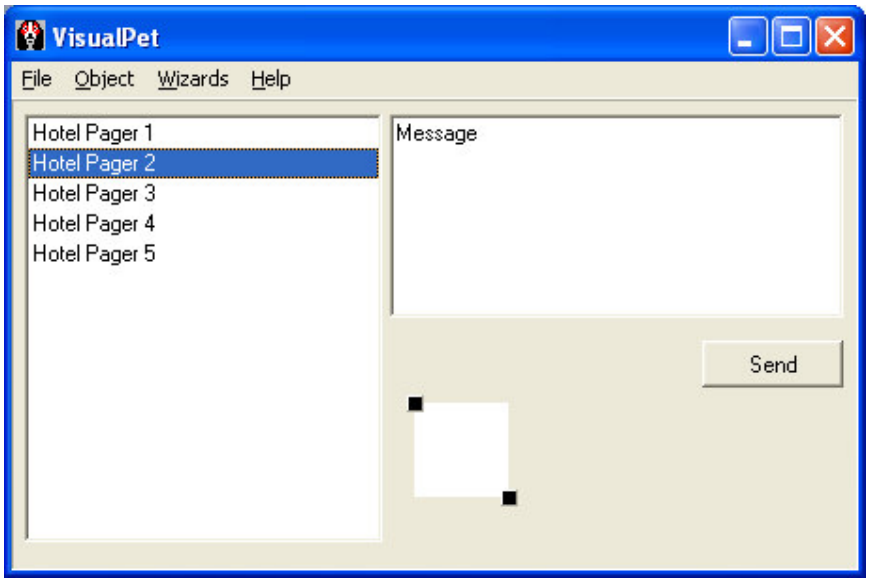
Return to the configuration window and Edit the "main" page again.
Use Object > New Image



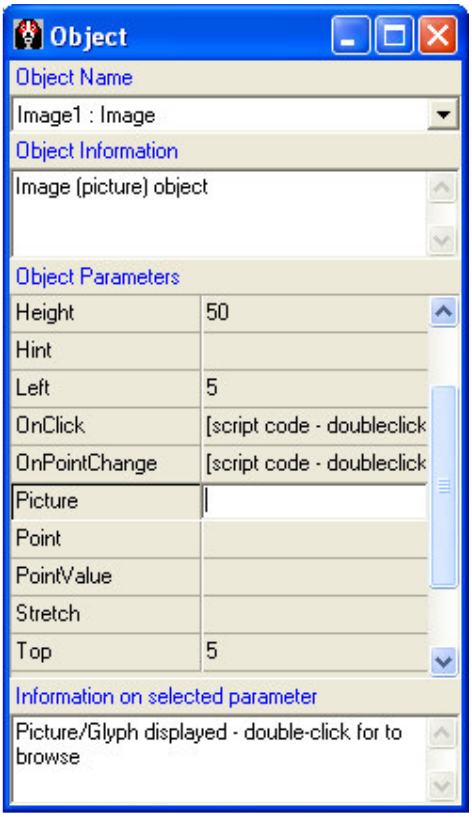

Leave Image1 and click "OK"



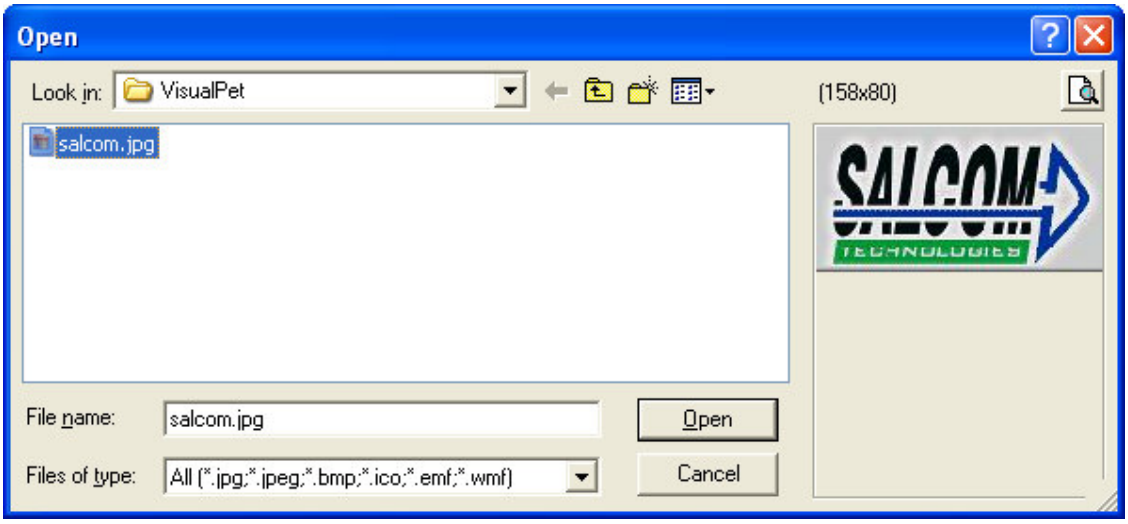
A small box will appear, use the mouse to move the box below the Message text box.



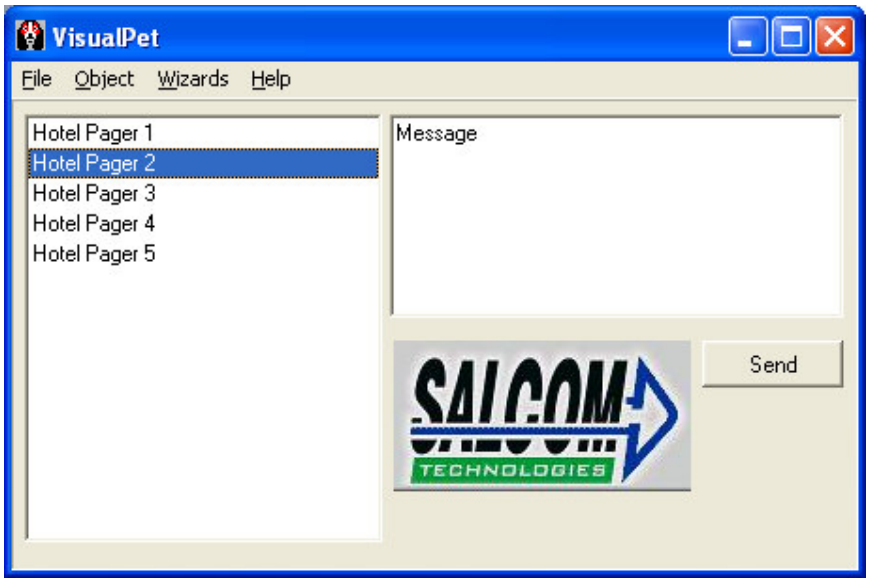

Inside the Object Inspector double click in area to the right of the Picture property.



This will open a file select window. You should see a file called Salcom.jpg, select this file and "Open"



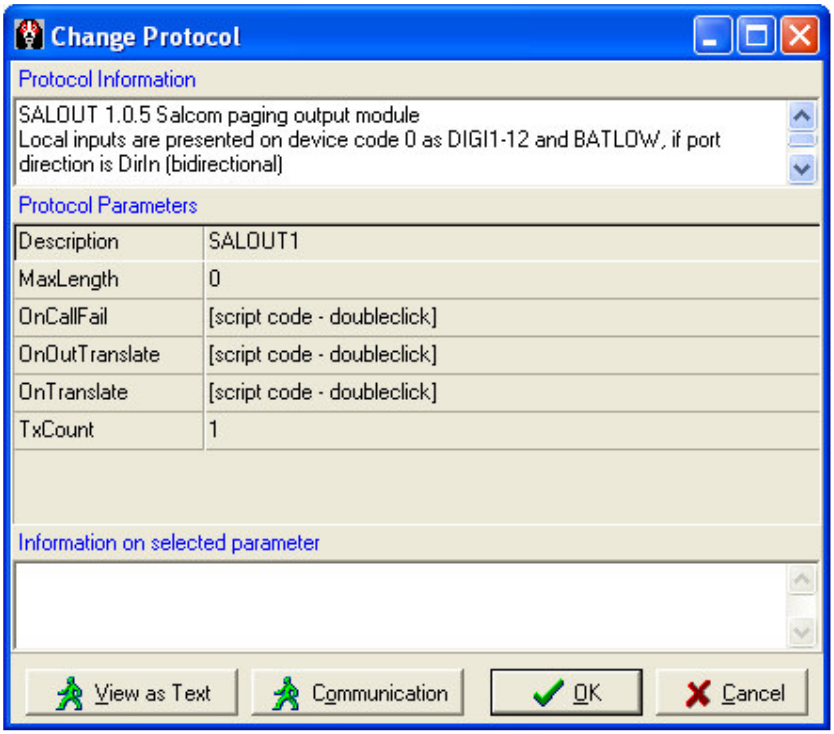

This will place the Salcom logo inside the "New Image" you placed. Resize and move the logo to fit beside the "Send" button.



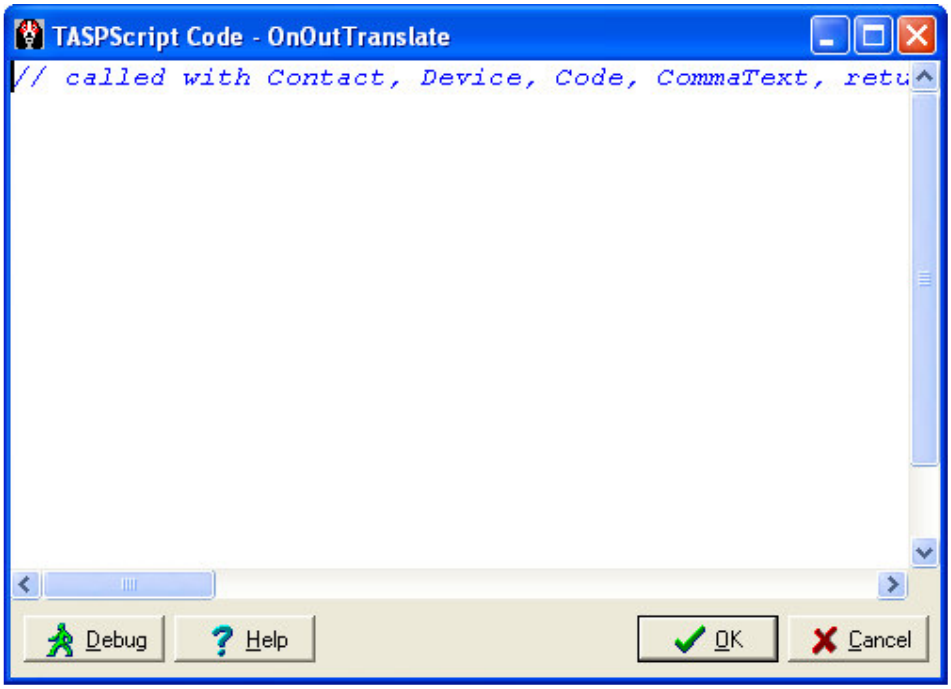

That's it, you have now added an image to your VisualPet configuration.

**Coding Example**

From the main window use File > View Router to open the router window, re-read Page 9 of this document to open the configuration of the Salout1 protocol.




Double click the [script code – doubleclick] to the right of the OnOutTranslate.

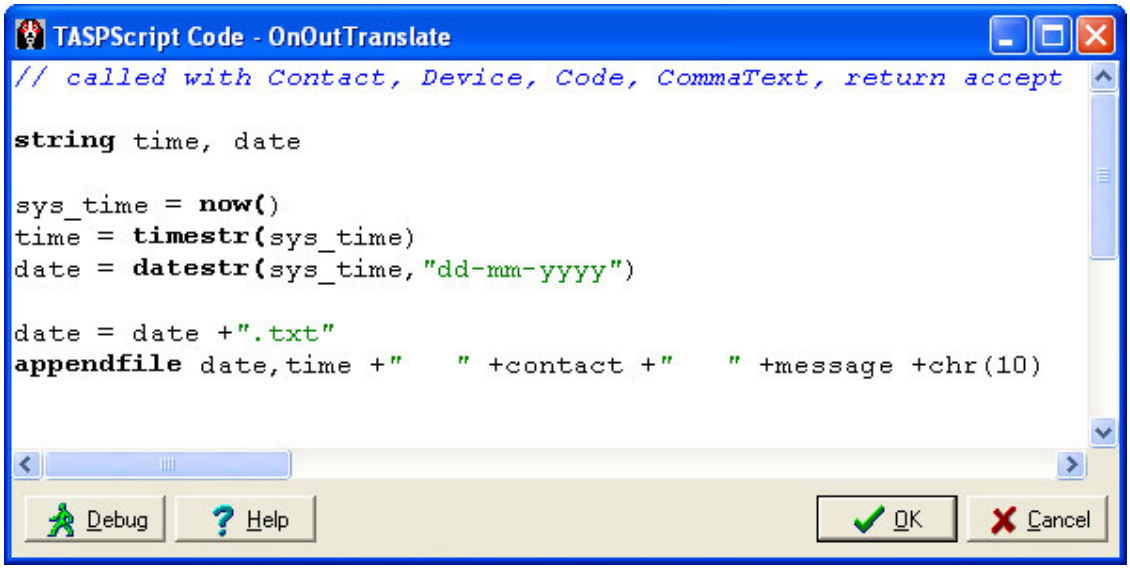


From here we will add some TASP script code to create a daily log file for all messages that we send using the SalOut1 protocol to the connected 12-62.

```
string time, date

sys_time = now()
time = timestr(sys_time)
date = datestr(sys_time,"dd-mm-yyyy")

date = date +".txt"
appendfile date,time +"   " +contact +"   " +message +chr(10)
```




Explaining the code we have now added:

string time, date
This declares some variables we wish to use for our script, both are of type "string", one called time the other called date.

sys_time = now()
This routine uses sys_time to get the actual time from the PC VisualPet is running on.

time = timestr(sys_time)
time is now set using the timestr() command to a string value of sys_time we can use.

date = datestr(sys_time,"dd-mm-yyyy")
datestr() uses only the date information in sys_time to create a string called date with:
dd = day, mm = month, yyyy = year
eg: 20-09-2009 (20th June 2009)

date = date +".txt"
now we take the date string (example above 20-09-2009) and add a .txt to the end.
The example would now be 20-09-2009.txt

```
appendfile date,time +"   " +contact +"   " +message +chr(10)
```

appendfile
appends a string to a file, creating the file as necessary, if the file doesn't exist then appendfile will create it.

appendfile <filename>, <data>

<filename>
The file to store the data in

<data>
The string to store within the file

So for our example the "appendfile date" will append a file called "20-09-2009.txt"

The data will be:
time +"   " +contact +"   " +message +chr(10)

time (eg:11:23:05 11am :23 minutes: 05 seconds)

| | |
|---|---|
| +"  " | Adds two spaces after the time. |
| +contact | Adds the Pager we sent the message to. |
| +"  " | Adds two spaces after the contact. |
| +message | Adds the Message we sent |
| +chr(10) | Adds a Line Feed after our line of text so each message logged is on a new line. |

So for a "Test Message" to Pager 1 (contact =1234501) the <data> will be:

11:23:05 1234501 Test Message<lf>

Each subsequent message added to the daily file 20-09-2009.txt could be:

11:23:05 1234501 Test Message<lf>
11:26:13 1234502 Please Call Reception<lf>
11:45:44 1234503 Parcel has arrived from Salcom<lf>

The above example shows how configurable the VisualPet interface is, from here you can config and create an application interface to suit almost anything ! Have another look at Page 3 of this document to see interfaces created using VisualPet for specific user applications.

For a full listing of commands and support login to:
http://vprouter.salcom.co.nz/